

REMARKS

This Amendment is filed in response to the Office Action mailed July 8, 2010.
All objections and rejections are respectfully traversed.

Claims 1-11, 17-34, and 36-41 are currently pending.

No new claims have been added.

Claims 1, 17, 19-22, and 37 have been amended.

Interview Summary

Applicant would like to thank Examiner Morrison for conducting the Applicant Initiated Interview on August 27, 2010 and for helping to advance this Application closer to allowance. Generally, as will be elaborated upon in greater detail below, the issue discussed involved Applicant's claimed **removing the hashed value of the selected entry of the first data set from the hash table in response to determining that the hashed value of the selected entry of the first data set is in the hash table**. Specifically, Applicant discussed that Heckel is silent to such a limitation.

While Examiner initially agreed with Applicant, Examiner noted that a closer look at the prior art would be required to verify Applicant's contentions and that another search may be conducted. If a new search results in new art, Applicant respectfully requests that Examiner contact the undersigned attorney to discuss the art before issuing the next Office Action. Examiner is encouraged to contact the undersigned attorney with any questions.

Rejections Under 35 U.S.C. §102

At paragraph 3 of the Office Action, claims 22-24, 26, and 40 were rejected under 35 U.S.C. §102(b) as being anticipated by Heckel (*A Technique for Isolating Differences Between Files*, Communications of the ACM, Volume 21, Number 4, April 1978) (hereinafter "Heckel").

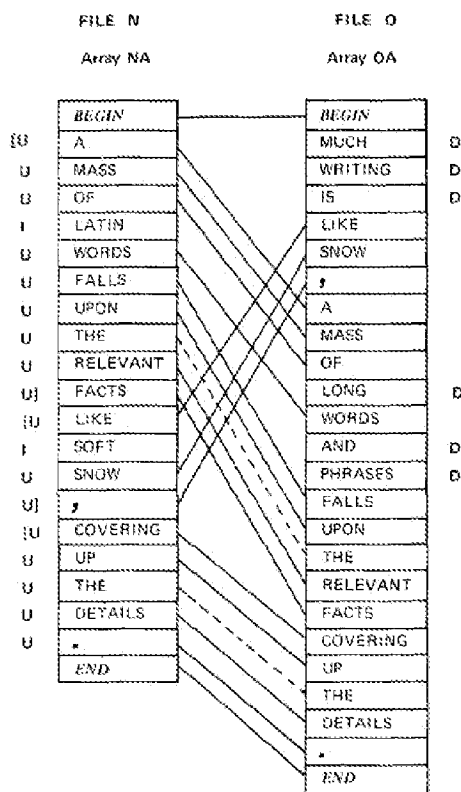
Applicant's claimed novel invention, as set forth in representative claim 22, comprises:

22. A computer method for comparing a first data set with a second data set, comprising:
- (a) selecting an entry from the first data set, the first data set stored on a source storage system;
 - (b) determining if a hashed value of the selected entry of the first data set is in a hash table, the hash table comprising one or more hashed values of the first data set;
 - (c) adding, in response to determining that the hashed value of the selected entry of first data set is not in the hash table, the hashed value of the selected entry of the first data set to the hash table;
 - (d) **removing from the hash table, in response to determining that *the hashed value of the selected entry of the first data set is in the hash table, the hashed value of the selected entry of the first data set***;
 - (e) selecting an entry from the second data set, the second data set stored on a destination storage system;
 - (f) determining if a hashed value of the selected entry of the second data set is in the hash table, the hash table further comprising one or more hashed entries of the second data set;
 - (g) adding, in response to determining that the hashed value of the selected entry of the second data set is not in the hash table, the hashed value of the selected entry of the second data set to the hash table;
 - (h) removing from the hash table, in response to determining that the hashed value of the selected entry of the second data set is in the hash table, the hashed value of the selected entry of the second data set;
 - (i) continuing (a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed; and
 - (j) reporting a difference between the first data set and the second data set in response to at least one hashed value remaining in the hash table.

Heckel teaches an algorithm for comparing two files by (visually) flagging and marking a “new” file (FILE N) and an “old” file (FILE O) (page 265, section 3, paragraphs 1-4). For example, Heckel states the following:

A formal version of this algorithm is presented in an earlier version of this paper [5]. It outputs each line flagged by the type of change it finds: [I]nsert, [D]elete, beginning of block, end of block, or [U]nchanged. (page 266, last paragraph before section 4) (emphasis added)

Fig. 1. Difference isolation. The hashcodes for the lines in the files being compared are the corresponding entries of arrays OA and NA. The first and last locations are used for unique generated begin and end lines. In pass 3, all unique lines are connected (solid lines). In passes 4 and 5, identical but nonunique lines are found (dashed lines). The file comparison can then be generated. Each inserts (I), delete (D), unchanged line (U), or moved block ([and]) can be detected and printed out. The output could be printed as: (page 266) (emphasis added)



Applicant respectfully contends that Heckel fails to anticipate Applicant's claimed novel **removing the hashed value of the selected entry of the first data set from the hash table in response to determining that the hashed value of the selected entry of the first data set is in the hash table.**

Applicant claims, in part, selecting an entry from a first data set and then determining if a hashed value of that selected entry is in a second hash table, the second hash

table comprising one or more hashed values from a second data set. In other words, broadly speaking, Applicant claims hashing entries of a first data set, storing one or more of the hashed values of the hashed entries in a first hash table, and then determining if a particular hashed value of the first data set in the first hash table matches one of those stored hash values located in the second hash table. **In response to determining that *the hashed value of the selected entry of the first data set is in the* (second) hash table,** Applicant claims **removing *the hashed value of the selected entry of the first data set from the hash table***. As the procedure progresses, **removing *the hashed value from the hash table*** in response to a match may advantageously result in a progressively faster algorithm reducing the impact on the processor.

Applicant respectfully contends that Heckel fails to anticipate Applicant's claimed novel **removing *the hashed value of the selected entry of the first data set from the hash table in response to determining that *the hashed value of the selected entry of the first data set is in the hash table****. Specifically, page 3 of the Office Action states, in part:

(d) removing from the hash table, in response to determining that the hashed value of the selected entry of the first data set is in the hash table, the hashed value of the selected entry of the first data set; (pass 5, page 266, section 4, third paragraph; figure 1; entry marked as match leaves differences unmarked which is equivalent of removal since it is removed from consideration) (emphasis added)

Put another way, the Office Action contends that Applicant's claimed novel **removing *the hashed value from the hash table*** is equivalent to removing an entry from consideration. However, Applicant respectfully disagrees, *inter alia*, for the following reasons:

1. The Office Action mistakenly states that any unmatched differences between two files are unmarked. However, Heckel unequivocally teaches marking each entry, regardless of finding a difference or not finding a difference, in the above citations and Fig. 1. Specifically, Heckel teaches that the file comparison algorithm marks each entry as: inserts (I),

delete (**D**), unchanged line (**U**), or moved block ([**and**]). Thus, Heckel does not leave differences unmarked.

2. To further expound on reason 1, because all entries are marked in some way, none of the entries are removed from consideration as suggested by the Office Action. For example, as can be seen from Heckel's description of Fig. 1, it is only after determining which entries should be marked with their appropriate flag that the file comparison may then be generated. In other words, because each marking is required to generate the file comparison, none of the entries are in fact removed from consideration. Furthermore, if any of the entries were in fact removed from consideration, then Heckel's algorithm would likely produce false positives by removing from consideration entries that correctly appear more than once (i.e., common articles such as the words "the" and "a").

3. To further expound on reason 2, removing a value from consideration is not the same as actually **removing the hashed value from the hash table**. For example, even if an entry is "removed from consideration", none of the marked entries are actually removed from the table, a point which is conceded by the Office by using an equivalence justification. Thus, as all entries remain in the table, Heckel's algorithm must still walk through each table entry, whether "removed from consideration" or not, in order to generate the file comparison. The necessity to determine similarities, mark all entries appropriately, and then additionally read each mark of each table entry requires more system resources. In contrast, as Applicant's claimed procedure progresses, **removing the hashed value from the hash table** in response to a match advantageously results in a progressively faster algorithm reducing the impact on the processor. Because Heckel does not actually remove an entry from the table, Heckel would not experience a reduced impact on the processor.

4. Lastly, Applicant respectfully directs the Office to **MPEP §2131**:

2131 Anticipation — Application of 35 U.S.C. 102(a), (b), and (e)

TO ANTICIPATE A CLAIM, THE REFERENCE MUST TEACH EVERY ELEMENT OF THE CLAIM

“A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)...(emphasis added)

See also **MPEP §2131.02**. “The identical invention must be shown in as complete detail as is contained in the... claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (emphasis added)

See also *Abbott Laboratories v. Sandoz, Inc.*, 500 F. Supp. 2d 807 - Dist. Court, ND Illinois, Eastern Div. 2007 citing *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548 (Fed. Cir. 1983) (stating that a prior art reference – in order to anticipate under 35 U.S.C. §102-must not only disclose all elements of the claim within the four corners of the document, but must also disclose those elements ‘arranged as in the claim.’ The test is thus more accurately understood to mean ‘arranged or combined in the same way as in the claim.’ *Id.* (emphasis added)

See also *Abbott Laboratories v. Sandoz, Inc.*, 500 F. Supp. 2d 807 - Dist. Court, ND Illinois, Eastern Div. 2007 citing *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1458-9 (Fed. Cir. 1984) (Reversing the district court’s finding of anticipation because the 102 reference ‘disclose[d] an entirely different device, composed of parts distinct from those of the claimed invention, and operating in a different way...’ further noting the reference was deficient because it did not disclose the elements of the claimed invention ‘arranged as in the claim’ as required by 35 U.S.C. §102.’ (emphasis added)

Thus, according to **MPEP §2131** and **MPEP §2131.02**, *inter alia*, even if “removing from consideration” were “equivalent” to Applicant’s claimed novel **removing the hashed value of the selected entry of the first data set from the hash table**, this is not enough to establish *prima facie* anticipation as required under 35 U.S.C. §102. Accordingly, because Heckel fails to teach each and every element as set forth in the claim, either expressly or inherently described as required under 35 U.S.C. §102, Heckel fails to anticipate Applicant’s claimed novel **removing the hashed value of the selected entry of the first data set from the hash table in response to determining that the hashed value of the selected entry of the first data set is in the hash table**.

Based on the foregoing, *inter alia*, Applicant respectfully contends that Heckel fails to anticipate the claimed invention under 35 U.S.C. §102 because of the absence from Heckel of Applicant's claimed novel **removing *the hashed value of the selected entry of the first data set from the hash table in response to determining that the hashed value of the selected entry of the first data set is in the hash table.***

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art was derived, at least in part, from the following excerpt(s):

Heckel

A formal version of this algorithm is presented in an earlier version of this paper [5]. It outputs each line flagged by the type of change it finds: [I]nsert, [D]elete, beginning of block, end of block, or [U]nchanged. (page 266, last paragraph before section 4) (emphasis added)

Fig. 1. Difference isolation. The hashcodes for the lines in the files being compared are the corresponding entries of arrays OA and NA. The first and last locations are used for unique generated begin and end lines. In pass 3, all unique lines are connected (solid lines). In passes 4 and 5, identical but nonunique lines are found (dashed lines). The file comparison can ***then*** be generated. Each inserts (**I**), delete (**D**), unchanged line (**U**), or moved block ([**and**]) can be detected and printed out. The output could be printed as: (page 266) (emphasis added)

Rejections Under 35 U.S.C. §103

At paragraph 5 of the Office Action, claims 1-11, 17-21, 25, 27-34, 36-39, and 41 were rejected under 35 U.S.C. §103(a) as being obvious over Heckel, in view of Pohlen, U.S. Patent Application Publication No. 3006/0015391 published on January 20, 2005 (hereinafter "Pohlen").

Applicant's claimed novel and non-obvious invention, as set forth in representative claim 1, comprises in part:

1. A computer method for comparing a first directory comprising unique elements with a second directory comprising unique elements, comprising:

- (a) for each entry in the first directory, placing a hash value of the entry in a hash table, the first directory stored on a source storage system;
- (b) selecting an entry of the second directory, the second directory stored on a destination storage system;
- (c) looking up a match between a hash value of the selected entry and the hash value of the entry in the hash table;
- (d) **removing, in response to the match between *the hash value of the selected entry and the hash value of the entry in the hash table, the hash value of the entry from the hash table***;
- (e) determining if an additional second directory entry exists;
- (f) looping to step (b) in response to identifying the additional second directory entry; and
- (g) reporting a difference between the first directory and the second directory in response to at least one hash value entry remaining in the hash table.

Pohlan teaches displaying a comparison result of at least two data structures organized in respective directory trees on a graphical display to allow a user to quickly and reliably grasp the differences between the directories [0011]. Notably, Pohlan is silent to Applicant's claimed source and/or destination *storage system*. However, even if it were assumed *arguendo* that such a limitation were shown, Pohlan is still silent to Applicant's claimed novel and non-obvious **removing, in response to the match between *the hash value of the selected entry and the hash value of the entry in the hash table, the hash value of the entry from the hash table***.

Additionally, as noted above with regard to claim 22, Heckel also fails to teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value of the selected entry of the first data set from the hash table in response to determining that the hashed value of the selected entry of the first data set is in the hash table***. As such, *inter alia*, because claim 1 comprises similar limitations of claim 22 not shown or suggested by any prior art reference (or any combination thereof), Applicant respectfully contends that Heckel, taken singly or in any combination with Pohlan, fails to render the presently claimed invention obvious under 35 U.S.C. §103(a). Specifically, Heckel and/or Pohlan, taken singly or in any combination, do not disclose, teach or suggest Applicant's claimed novel and non-obvious **removing, in response to the match between**

the hash value of the selected entry and the hash value of the entry in the hash table, the hash value of the entry from the hash table.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art was derived, at least in part, from the following excerpt(s):

Pohlan

[0011] Thus, an object of the present invention is to provide a method as well as a data processing system for visualizing a comparison result of at least two data structures organized in comparison trees, which enables a clear display of the comparison result that the user can quickly and reliably grasp.

Conclusion

All new claims and/or claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Michael T. Abramson/
Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927